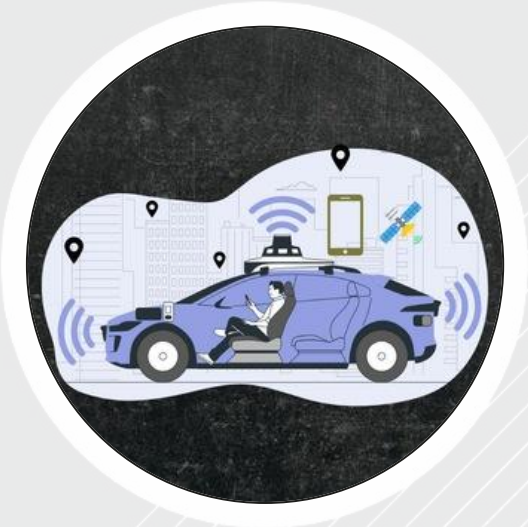
SPYWOLF

Security Audit Report

(TESTNET)



Completed on

November 1, 2023

OVERVIEW

This audit has been prepared for **VANETCHAIN** to review the main aspects of the project to help investors make make an informative decision during their research process.

You will find a a summarized review of the following key points:

- ✔ Contract's source code
- ✔ Owners' wallets
- ✔ Tokenomics
- ✔ Team transparency and goals
- ✔ Website's age, code, security and UX
- ✔ Whitepaper and roadmap
- ✔ Social media & online presence

The results of this audit are purely based on the team's evaluation and does not guarantee nor reflect the projects outcome and goal

- SPYWOLF Team -

TABLE OF CONTENTS

VANETCHAIN

PROJECT DESCRIPTION

According to their whitepaper:

VANETC is the token of VanetChain on BSC Network. VanetChain aims to make a custom and private blockchain to secure Vehicle Ad-hoc Networking. Vehicle Ad hoc networking is the main transmission medium for smart vehicle communication. Because of the security issues that can cause catastrophic results like accidents and human death, A secure implementation of VANET is crucial. To handle the security issues and make a safe network for vehicle communication we provide a blockchain based vehicle messaging framework.

Release Date: Presale starts in November, 2023

Category: Wallet

CONTRACT INFO

Token Name
Test-v

Symbol
T-v

Contract Address
0x5192650b3a2eD3314788b2e00AcFba5A51f9ab6D

Network
Sepolia testnet

Language
Solidity

Deployment Date
Oct 31, 2023

Contract Type
Staking

Total Supply
500,000,000

Status
Not launched

TAXES

Buy Tax
7%

Sell Tax
5%

*Additional information about taxes here Additional information about taxes here Additional information about taxes here

Our Contract Review Process

The contract review process pays special attention to the following:

- Testing the smart contracts against both common and uncommon vulnerabilities
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Blockchain security tools used:

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

CURRENT STATS

(As of June 02, 2022)

Liquidity

Not added yet

Burn

No burnt tokens

Status:

Not Launched!

MaxTxAmount

1,500

Please Add additional info here

LP Address(es)

Liquidity not added yet

TOKEN TRANSFERS STATS

Transfer Count	TESTNET
Uniq Senders	TESTNET
Uniq Receivers	TESTNET
Total Amount	TESTNET
Median Transfer Amount	TESTNET
Average Transfer Amount	TESTNET
First transfer date	TESTNET
Last transfer date	TESTNET
Days token transferred	TESTNET

SMART CONTRACT STATS

Calls Count	TESTNET
External calls	TESTNET
Internal calls	TESTNET
Transactions count	TESTNET
Uniq Callers	TESTNET
Days contract called	TESTNET
Last transaction time	TESTNET
Created	TESTNET
Create TX	TESTNET
Creator	TESTNET

FEATURED WALLETS

Owner address	TESTNET
Marketing fee receiver	TESTNET
LP address	TESTNET

TOP 3 UNLOCKED WALLETS

1 1.17% TESTNET

2 1.01% TESTNET

3 0.77% TESTNET

VULNERABILITY CHECK

Design Logic	Passed
Compiler warnings.	Passed
Private user data leaks	Passed
Timestamp dependence	Passed
Integer overflow and underflow	Passed
Race conditions and reentrancy. Cross-function race conditions	Passed
Possible delays in data delivery	Passed
Oracle calls	Passed
Front running	Passed
DoS with Revert	Passed
DoS with block gas limit	Passed
Methods execution permissions	Passed
Economy model	Passed
Impact of the exchange rate on the logic	Passed
Malicious Event log	Passed
Scoping and declarations	Passed
Uninitialized storage pointers	Passed
Arithmetic accuracy	Passed
Cross-function race conditions	Passed
Safe Zeppelin module	Passed
Fallback function security	Passed

THREAT LEVELS

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Low Risk

Issues on this level are minor details and warning that can remain unfixed.

Informational

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

FOUND THREATS

⚠ High Risk

Owner can enable/disable staking. When staking is disabled, new stakes will not be accepted and the already staked users won't be able to withdraw their current stakes.

```
function EnableStake(uint256 enable) public onlyOwner{
    uint256 state = enable;
    setStakeState(state);
}

function withdrawStake(uint256 _duration)  public {
    uint256 burnt_amount=0;
    (uint256 _amount_to_unstake,uint256 burnt_flag,uint256 durtn) = _withdrawStake(_duration);
    ..................
}

function _withdrawStake(uint256 _duration) internal returns(uint256,uint256,uint256){
    require(stake_enabled==1,"Stake must be enabled");
.............
}
```

- Recommendation:
  - Staking disable should affect only new users, but not ones that are already staked.

FOUND THREATS

⚠ High Risk

Owner can burn tokens from the contract and owner's wallet.
If inappropriate amount of tokens are burnt while users are staking, users won't be able to withdraw their staked tokens.

```
function burn(address account, uint256 amount) public onlyOwner returns(bool) {

require(account==address(this) || account==owner(),"Burn address must be token adresss or owner address");
    _burn(account, amount);
    return true;
}

function _burn(address account, uint256 amount) internal {
    require(account != address(0), "Can not burn from the zero address");
    require(_balances[account] >= amount, "Can not burn more than the account owned tokens.");

    _balances[account] = _balances[account] - amount;
    _totalSupply = _totalSupply - amount;
    emit Transfer(account, address(0), amount);
}
```

- Recommendation:
  - Any manual burns should not affect the already staked tokens in the contract.

FOUND THREATS

⚠ High Risk

If user's withdraw amount (staked+rewards) is higher than the contract's current token balances, transfer will fail and user will be unable to unstake their tokens.

```
function withdrawStake(uint256 _duration)  public {
    uint256 burnt_amount=0;
    (uint256 _amount_to_unstake,uint256 burnt_flag,uint256 durtn) = _withdrawStake(_duration);
    return ((current_stake.amount+reward),stake_burnt_flag,duration);

    if(burnt_flag==1){
        if(durtn==15){
            burnt_amount= _amount_to_unstake / 100;
        }
        else{
            burnt_amount = _amount_to_unstake / 80;
        }

        _amount_to_unstake = _amount_to_unstake - burnt_amount;
        _burn(address(this),(burnt_amount*(10**18)));
    }
    _transfer(address(this),msg.sender,(_amount_to_unstake*(10**18)));
}

function calculateStakeReward(Stake memory _current_stake,uint256 s_duration) internal view returns(uint256){
    uint256 annual_rate = 0;

    if(_current_stake.duration == 15){
        annual_rate = 12;
    } else{
        annual_rate=16;
    }

    if(_current_stake.amount == 0){
      return 0;
    }

    return (annual_rate *_current_stake.amount * s_duration)  / (100*365*24);
}

function _withdrawStake(uint256 _duration) internal returns(uint256,uint256,uint256){
.............................................
    Stake memory current_stake = stakeholders[user_index].address_stakes[duration_map[_duration]];
    uint256 reward = calculateStakeReward(current_stake,stake_duration);
    totalStaked = totalStaked - current_stake.amount;
    delete stakeholders[user_index].address_stakes[duration_map[_duration]];
    return ((current_stake.amount+reward),stake_burnt_flag,duration);
}

function _transfer(address sender, address recipient, uint256 amount) internal {
    require(sender != address(0), "Do nottransfer from zero address");
    require(recipient != address(0), "Do not transfer to zero address");
    require(_balances[sender] >= amount, "Can not transfer more than you own");

    _balances[sender] = _balances[sender] - amount;
    _balances[recipient] = _balances[recipient] + amount;

    emit Transfer(sender, recipient, amount);
}
```

- Recommendation:
    - Any additional tokens (rewards) should be taken into consideration on withdrawal calculations.

Informational

Users can stake their tokens for Annual Percentage Yield (APY) of 12% (for 15 days staking) and 16% (for 30 days staking).

```
function stake(uint256 _amount, uint256 _duration) public {

    require(_amount*(10**18) < _balances[msg.sender], "Cannot stake more than you own");
    _stake(_amount, _duration);
    _transfer(msg.sender,address(this),_amount*(10**18));
}

function _stake(uint256 _amount, uint256 _duration) internal{
    uint256 amount= _amount;
    require(stake_enabled==1,"Stake must be enabled");
    require(amount >= 100000 && amount<=2000000, "Cannot stake , change amount");
.............
}

function calculateStakeReward(Stake memory _current_stake,uint256 s_duration) internal view returns(uint256){
    uint256 annual_rate = 0;

    if(_current_stake.duration == 15){
        annual_rate = 12;
    } else{
        annual_rate=16;
    }

    if(_current_stake.amount == 0){
      return 0;
    }

    return (annual_rate *_current_stake.amount * s_duration)  / (100*365*24);
}
```

RECOMMENDATIONS FOR

GOOD PRACTICES

1. Consider fundamental tradeoffs
2. Be attentive to blockchain properties
3. Ensure careful rollouts
4. Keep contracts simple
5. Stay up to date and track development

VANETCHAIN

GOOD PRACTICES FOUND

- ✔ The owner cannot mint new tokens after deployment
- ✔ The owner cannot set a transaction limit

SPYWOLF

CRYPTO SECURITY

Audits | KYCs | dApps
Contract Development

ABOUT US

We are a growing crypto security agency offering audits, KYCs and consulting services for some of the top names in the crypto industry.

- **OVER 150 SUCCESSFUL CLIENTS**
- **MORE THAN 500 SCAMS EXPOSED**
- **MILLIONS SAVED IN POTENTIAL FRAUD**
- **PARTNERSHIPS WITH TOP LAUNCHPADS, INFLUENCERS AND CRYPTO PROJECTS**
- **CONSTANTLY BUILDING TOOLS TO HELP INVESTORS DO BETTER RESEARCH**

To hire us, reach out to
contact@spywolf.co or
t.me/joe_SpyWolf

FIND US ONLINE

SPYWOLF.CO

SPYWOLF.NETWORK

@SPYWOLFNETWORK

@SPYWOLFOFFICIAL

@SPYWOLFNETWORK

@SPYWOLFNETWORK

Disclaimer

This report shows findings based on our limited project analysis, following good industry practice from the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, overall social media and website presence and team transparency details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report.
While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER:
By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice.
No one shall have any right to rely on the report or its contents, and SpyWolf and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (SpyWolf) owe no duty of care towards you or any other person, nor does SpyWolf make any warranty or representation to any person on the accuracy or completeness of the report.
The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and SpyWolf hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, SpyWolf hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SpyWolf, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts, website, social media and team.
No applications were reviewed for security. No product code has been reviewed.